NASA-Ames Research Center

Human Systems Integration

Man-machine Integration Design and Analysis System (MIDAS) 4.0 (Beta) Training Class

Wednesday October 11, 2006

San José State University

MIDAS
man-machine integration design and analysis system

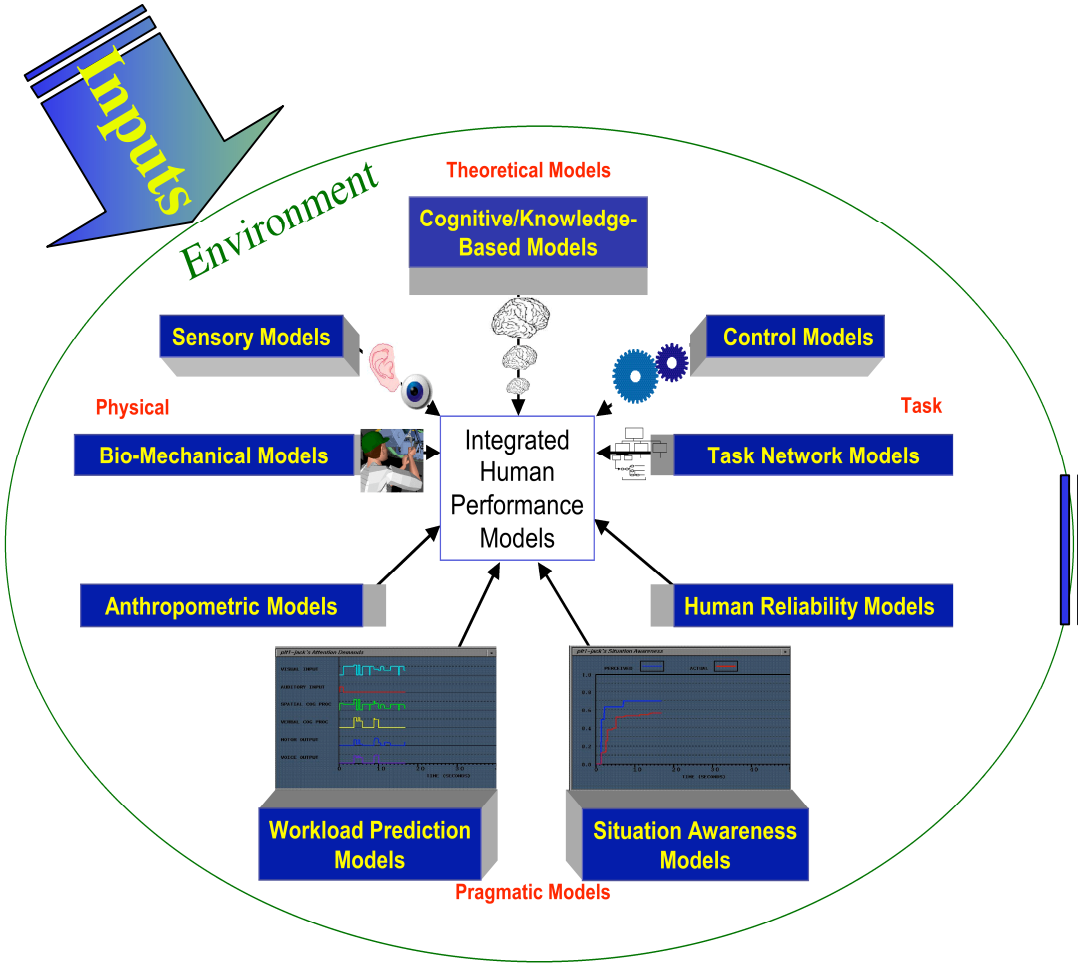
Attendees and Affiliations

- Shelly Scott-Nash - Alion Science and Technology (Micro Analysis and Design Operations)
- Dr. Ronald Boring - Idaho National Laboratory
- David Keller - Idaho National Laboratory
- Dr. Tuan Tran - Idaho National Laboratory
- Dr. Brent Buetter - NASA Ames Research Center
- Dr. David Foyle - NASA Ames Research Center
- Sandra Hart - NASA Ames Research Center
- Becky Hooey - SJSU/NASA Ames Research Center
- Brian Gore - SJSU/NASA Ames Research Center
- Ujwala Ravinder - SJSU/NASA Ames Research Center

Overview

- Human Performance Modeling
- MIDAS 4.0 Modeling
  - MIDAS 4.0 Platform
  - MIDAS 4.0 Components
  - MIDAS 4.0 Computational Structures
- MIDAS 4.0 Graphic User Interface/Environment
  - MIDAS Model Development Environment
  - Jack's Anthropometric and CAD Development Environment
  - MIDAS 4.0 Procedural GUI (Apex)
  - MIDAS 4.0 Output screens

Human Performance Model Components

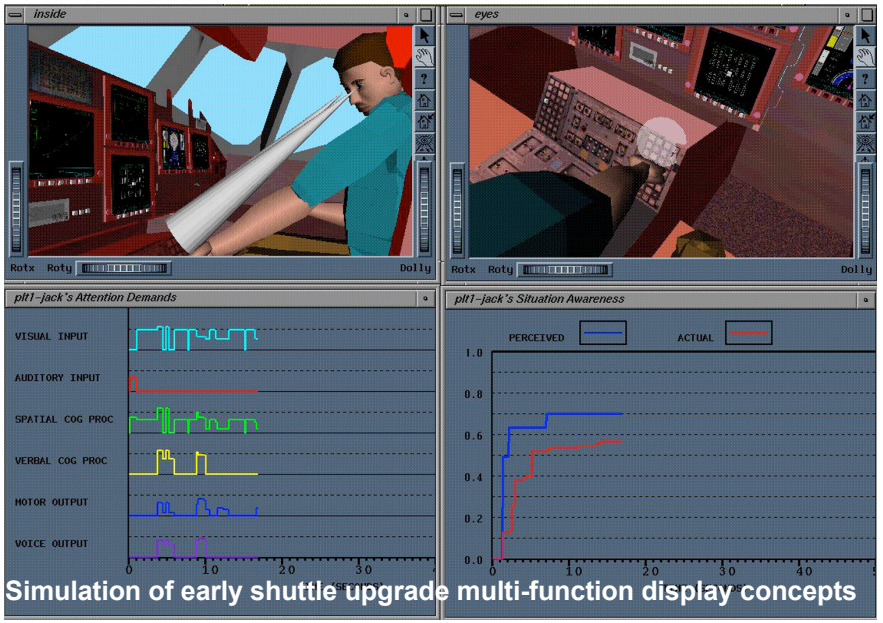


Outputs

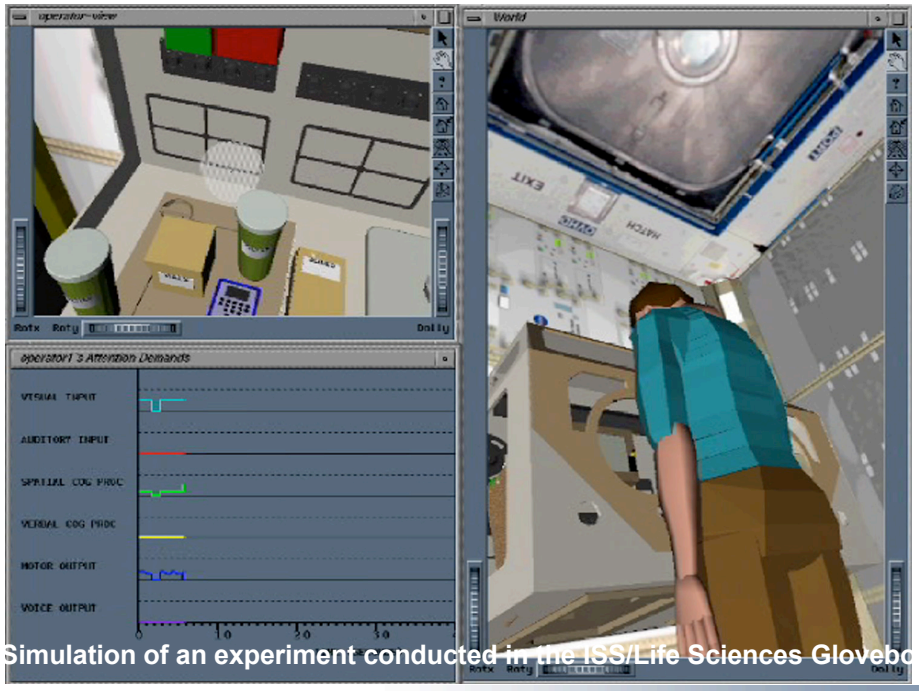
- Procedures, Activities, Timelines
- Agenda Management
- Performance: Workload
- Performance: Time
- Performance: Situation Awareness
- Performance: Human Errors
- Human and System Risks
- 2D and 3D Dynamic Visualization
- FoV/Reach Envelope

MIDAS

- ✓ Integrates, validated 1st principle models of human behavior with….
- ✓ ….3D CAD models of the environment, the workstation, and the equipment
- ✓ Produces task timelines, workload and situation awareness profiles



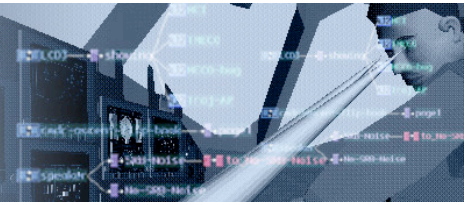
Simulation of early shuttle upgrade multi-function display concepts



Simulation of an experiment conducted in the ISS/Life Sciences Glovebo

- ✓ Hosted on a PC/Windows platform
- ✓ Generates realistic task-management behaviors sensitive to task context, environment
- ✓ Controls a generic, anthropometrically-correct human representation (Jack™, 5th percentile female - 95th percentile male)

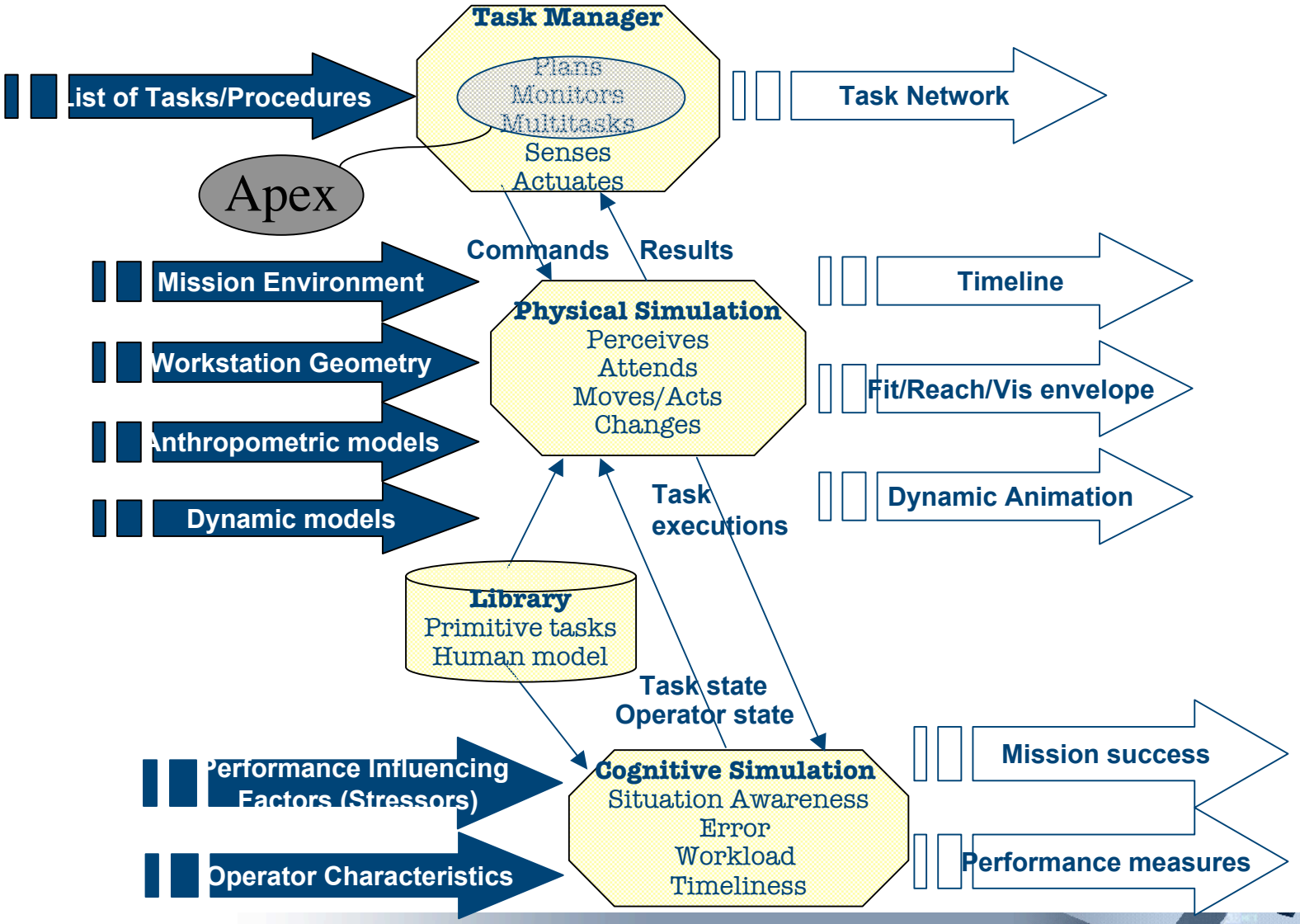
MIDAS
man-machine integration design and analysis system



NASA-Ames Research Center

Human Systems Integration

MIDAS Structures



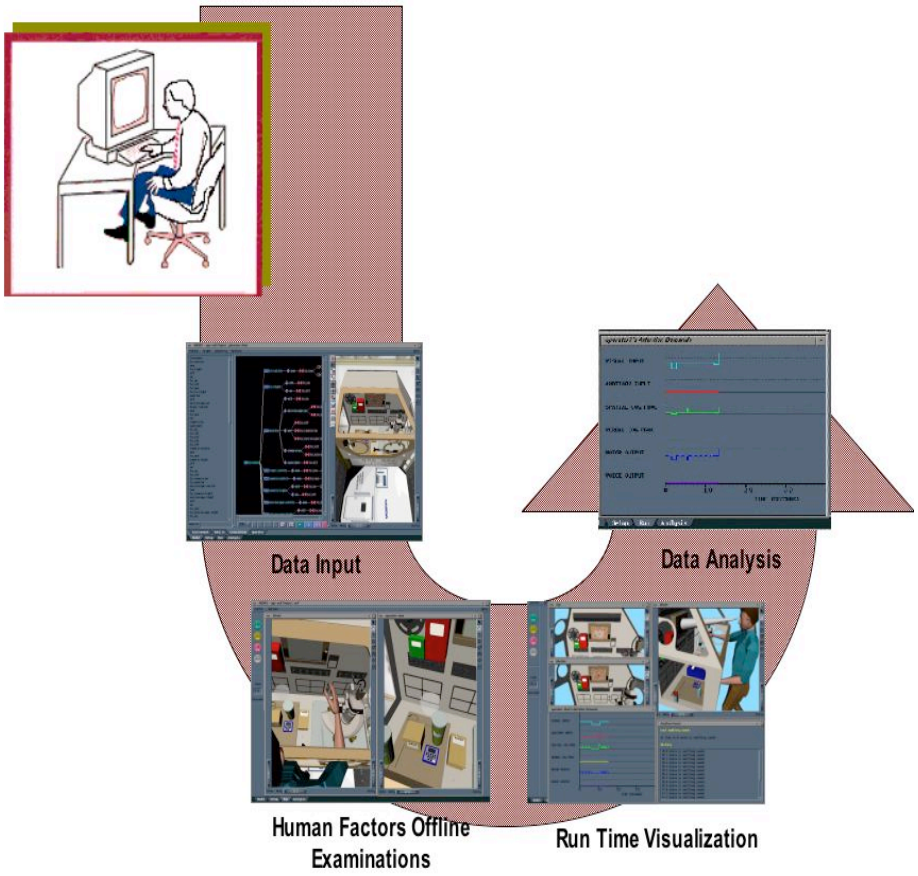
MIDAS

man-machine integration design and analysis system

NASA-Ames Research Center

Human Systems Integration

MIDAS Development Environment



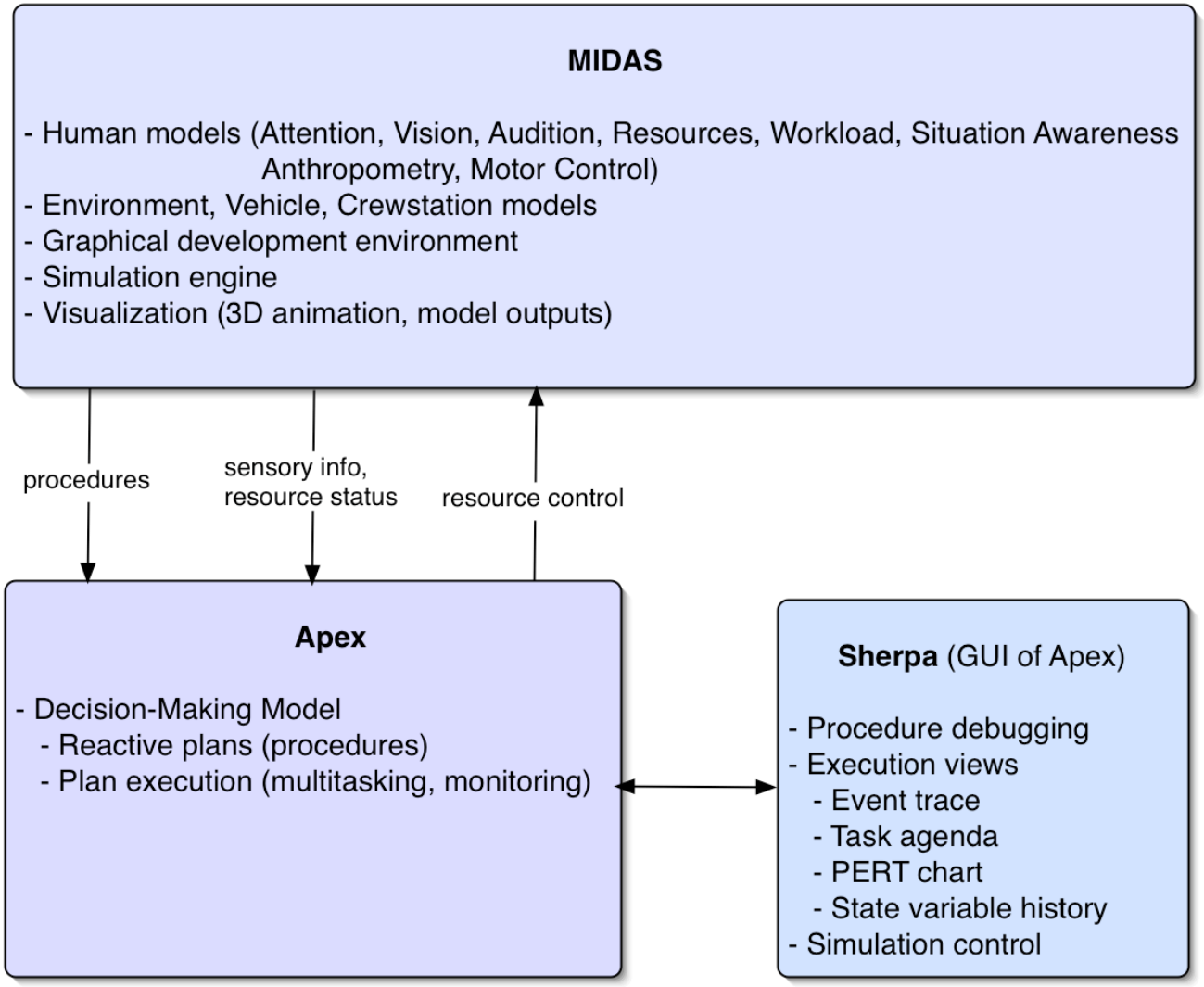
- 3-D rapid prototyping human performance modeling and simulation environment
- Produces quantitative predictions of human-system safety (resource use and allocation)
- Supports design of automation, crew stations and operating procedures
- Reduces design cycle time
- Top-down, bottom-up integrated structure allows generation of emergent behaviors
- Integrated desktop simulation environment allows users to:
  - Enter procedures/Populate the environment
  - Examine run-time visualization
  - Analyze workload (6-channel)
  - Analyze situation awareness
  - Performance error predictions
  - Examine human contribution to system risk
  - Conduct safety evaluations

NASA

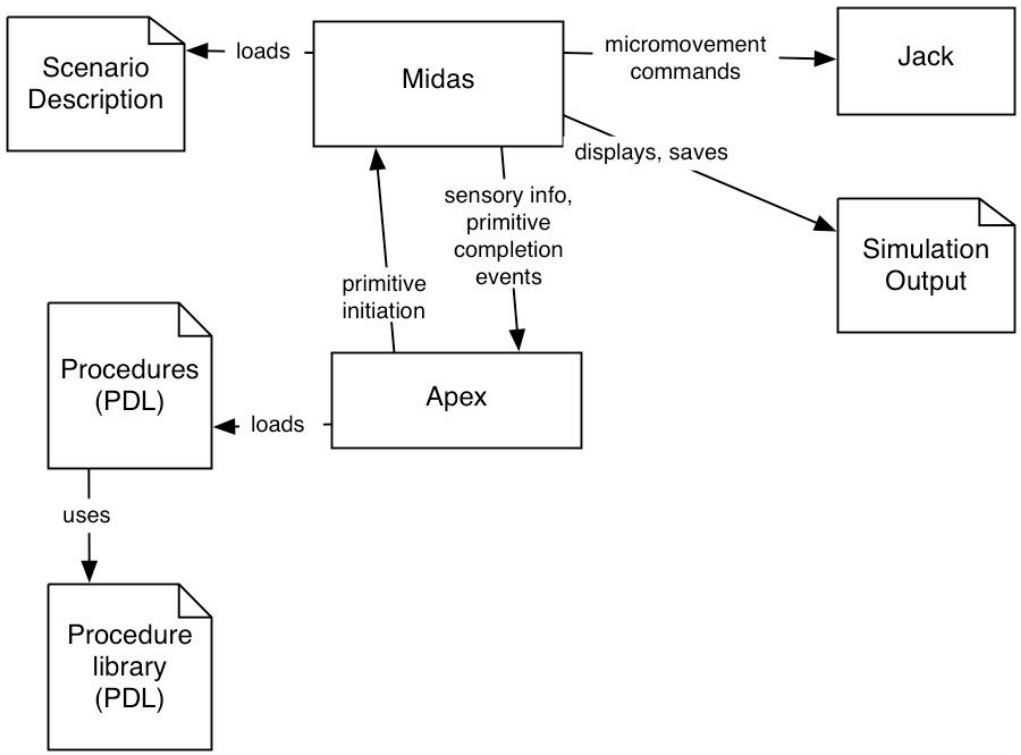
MIDAS

man-machine integration design and analysis system

MIDAS-Apex Structure



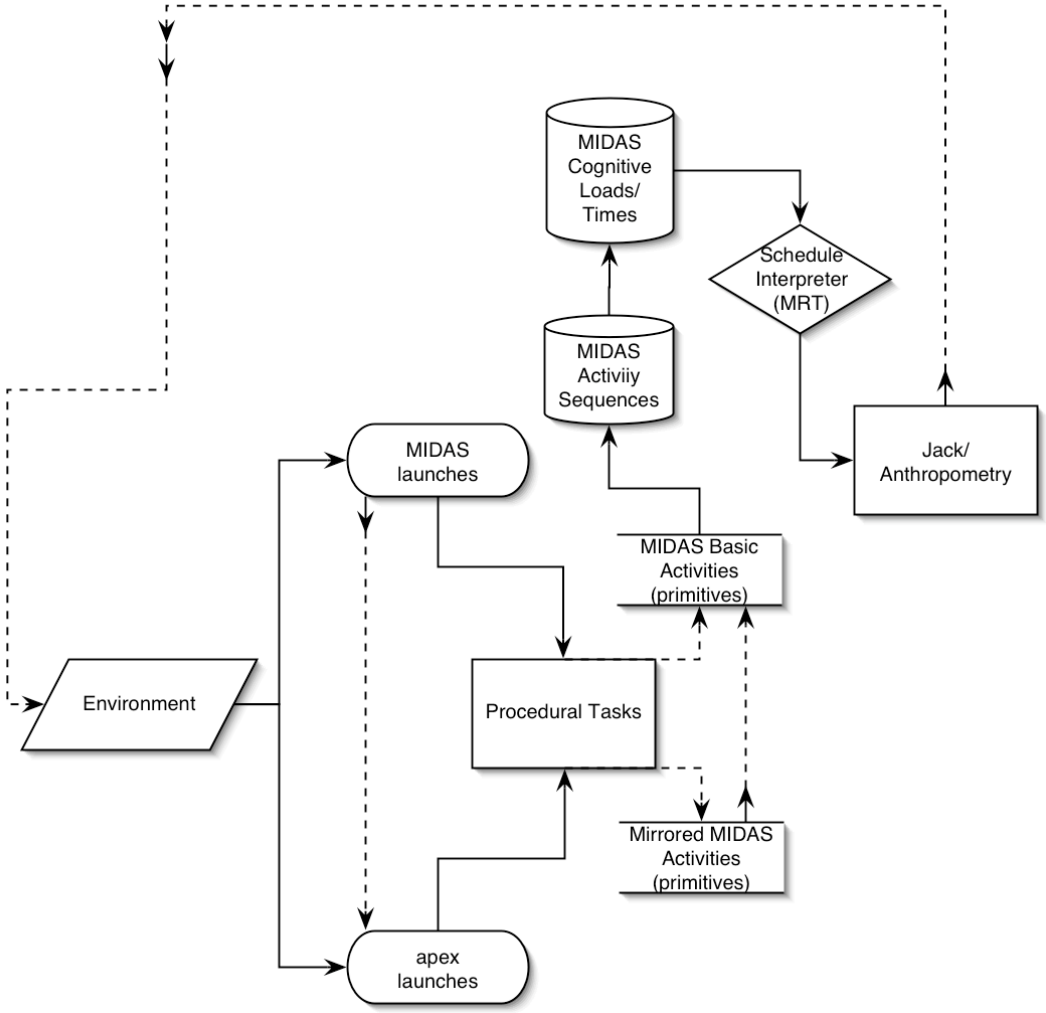
MIDAS Model Interaction



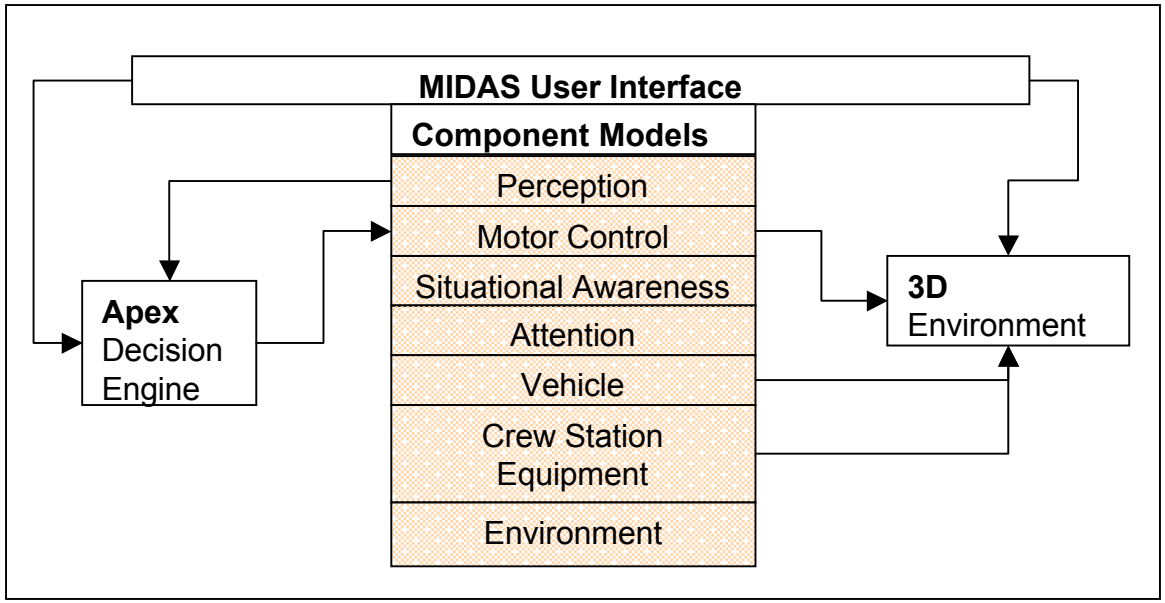
- Sequence of events (4-6 form a repeating loop)
  - Midas loads a specified scenario description (.scene)
  - Apex loads a corresponding procedure file for each Operator (a .lisp file having same name as the Operator, which might use other PDL library files -- that are also then loaded)
  - A procedure having the index *(do-domain)* must exist for each operator. This is the operator's initial task, which is initiated when the simulation is started
  - As the simulation runs, sensory information (vision, hearing etc) is sent to Apex by the Midas operator(s); Apex executes procedures, which either decompose into other procedures or initiate operator primitives in Midas.
  - Operator primitives initiated by Apex are "macro" level commands (e.g. reach and press the red button with left index finger). Midas orchestrates each primitive over a computed duration, while also computing workload and other simulation outputs. It sends Jack micro movement commands at each simulation tick.
  - When an operator finishes a primitive task, a signal is sent back to Apex, which allows further refinement of the operator's task agenda (e.g. additional procedures become enabled and run).

The MIDAS-Apex-Jack Structure

- Information passing among the component models in MIDAS 4.0



MIDAS Interface Communication Description



- Operator procedures and dependencies simulated in Apex.
- Dependencies are satisfied when the simulated operator perceives an event, not when the event occurs.
- The operator affects changes to the equipment through MIDAS primitive actions, not by changing the equipment attributes directly.
- MIDAS calculates duration for procedures, tracks operator attention (workload) and SA.

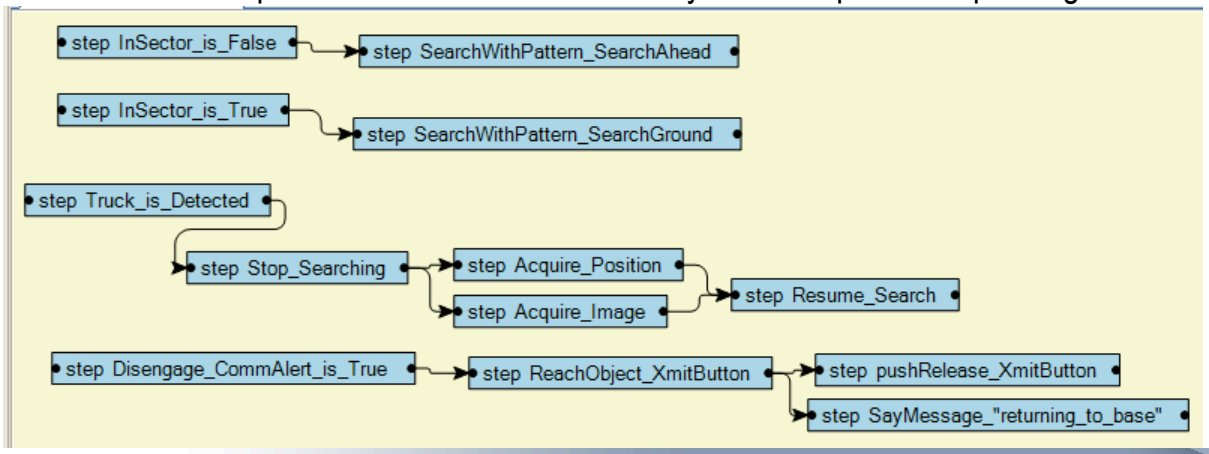
- Apex and the 3D Environment can be collocated on the same computer or different computers on the same network.
- 3D Environment used for visualization.

MIDAS 3D CAD Environment

- Implemented using UGS Classic Jack
  - Interface structured to support integration of alternate environments with moderate effort
  - Classic Jack provides us with the ability to integrate new CAD formats (Pro-E and JT) in addition to inventor formats
- Current functionality
  - Used for visualization of simulation
  - Wizard developed to match 3D Environment Elements to MIDAS Elements
    - Human figure names and effector initial locations
    - Vehicle names and waypoint paths
    - Equipment names, initial location and size
    - Environment feature names, location and size

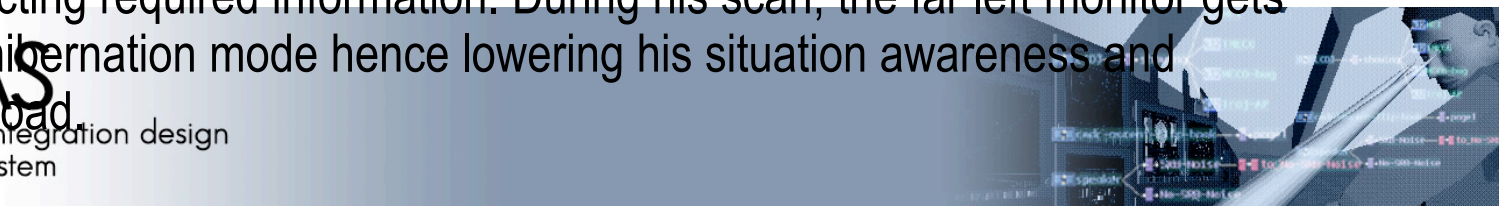
Enhanced User Interface

- Procedure Definition Language (PDL)
  - Apex is a powerful **computational** decision engine utilizing a LISP-derived procedure language
  - New GUI allows user to enter only the elements that make sense to both MIDAS and Apex without any need to program in LISP
    - query beliefs
      - Given a scan pattern, has the operator perceived that the warnings and alerts panel is visible?
    - query perception level
      - Has the operator dwelled long enough cumulatively on the dial to have attained Exact Read level of perception?
    - dependencies
      - The operator will wait until the auditory alarm stops before pushing the acknowledge button.



Scenario Description: Hands-on Project

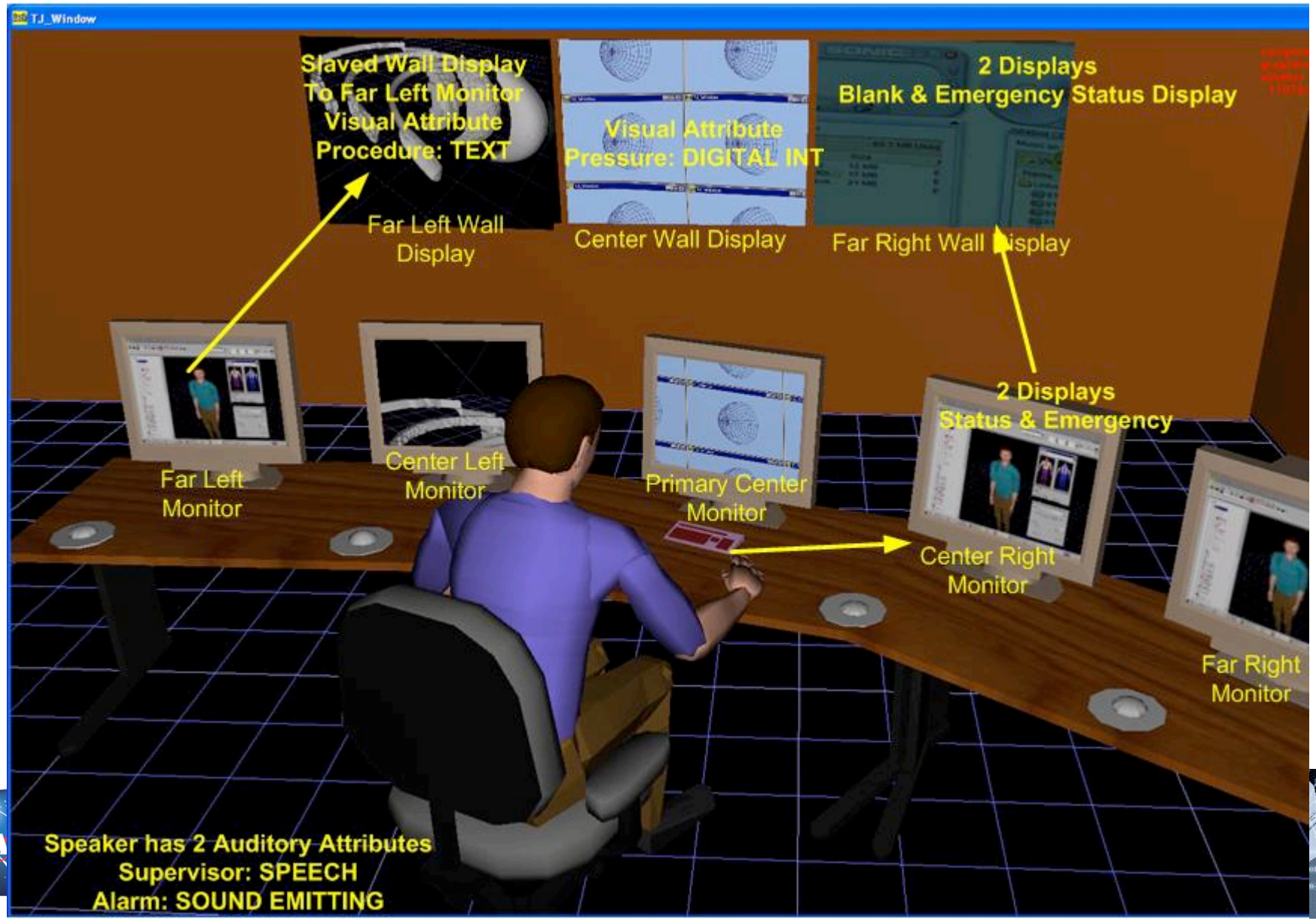
- Made up of a human operator in a control room, with 5 monitors and 3 wall screens that reflect the displays of the main monitors, a keyboard & 5 trackballs to interface with the monitors
- Operator's goal:
  - Responsible for monitoring the five displays and responding to both visual and auditory alerts in the simulation.
  - Begins with a nominal scan of the monitors
  - Responds to an auditory alarm by reaching the keypad with his left hand and acknowledging the alarm
    - This causes a change one of the monitors to display emergency information which is also displayed on the wall screen.
  - Receives a request from his supervisor to check pressure values. He looks up with an emergency scan pattern and checks for a pressure value over 80. He comprehends the information displayed on the screen extracting required information. During his scan, the far left monitor gets into hibernation mode hence lowering his situation awareness and workload.



NASA-Ames Research Center

Human Systems Integration

Scenario Description: Control Room CAD



MIDAS 4.0 Basics

- 4 development windows + Jack CAD environment
  - MIDAS Procedure development environment: “Apex Author”
  - MIDAS Properties View
  - MIDAS Tree View
  - MIDAS Output View
    - Shows the code running in a per-tick format
    - Workload - shows tick based workload output
    - Situational Awareness - shows tick based actual vs. perceived SA
    - Timeline information - shows tick based task performance
    - Task completion/schedule - success criteria
  - Jack CAD and anthropometry
- Window management - Remember that you can always change the working area of the MIDAS 4.0 windows to make it easier to develop simulations

MIDAS 4.0 - Basics

- A “+” sign beside an element in a list means that there is information about a relationship between two elements
- Hovering your mouse (cursor) over a location for text strings that are too long
  - Hovering is an attribute of the property grid
- Pause points - used to test/debug the simulation
- Tree view - Hierarchical equipment screen used for high level view of the structures in the simulation
- User will need to type information into string variables (the text attribute or say message)
- Right clicking the mouse will enable you to insert variables on a list
  - E.g. MIDAS procedure development (Apex Author) - steps/procedures in the sim.
- Edit properties in the properties window (grid)
- Collection editor is the editor for attributes, accessible via … ellipse button
- Simulations need to be defined after all lists are created
- Dependency, Dependency path node

MIDAS 4.0 - Basics

- MIDAS Primitives - basis of MIDAS behaviors, reduced from SGI version of MIDAS (rolled redundant ones together and we do MIDAS primitive selection based on attribute types)
  - Reach
  - Push-and-release
  - Visual - fixate object - breaks out into alphanumeric, spatial, color, default driven by attributes
  - Visual - scan with pattern
  - Stop scanning, interrupt scanning
  - Auditory - monitor, listen to speech (attribute dependent), listen to signal
  - Say-message

MIDAS 4.0 - Basics

- MIDAS Procedures
  - Made up of a series of Apex steps and Apex “wait-for” conditions
- Apex Steps
  - Sequence of activity (step 1 - reach, step 2 push-and-release, etc.)
  - Schedule MIDAS primitives and attribute tests

MIDAS 4.0 - Basics

- Simulation
- External connection
- Crewstation list
- Environment list
- Event list
- Operator list
- Situational Awareness list
- Vehicle list
- Apex procedure list

- MIDAS 4.0 Beta requires a definition in each of the classifications
- For this beta version of MIDAS, please define at least one for each

MIDAS 4.0 - Simulation

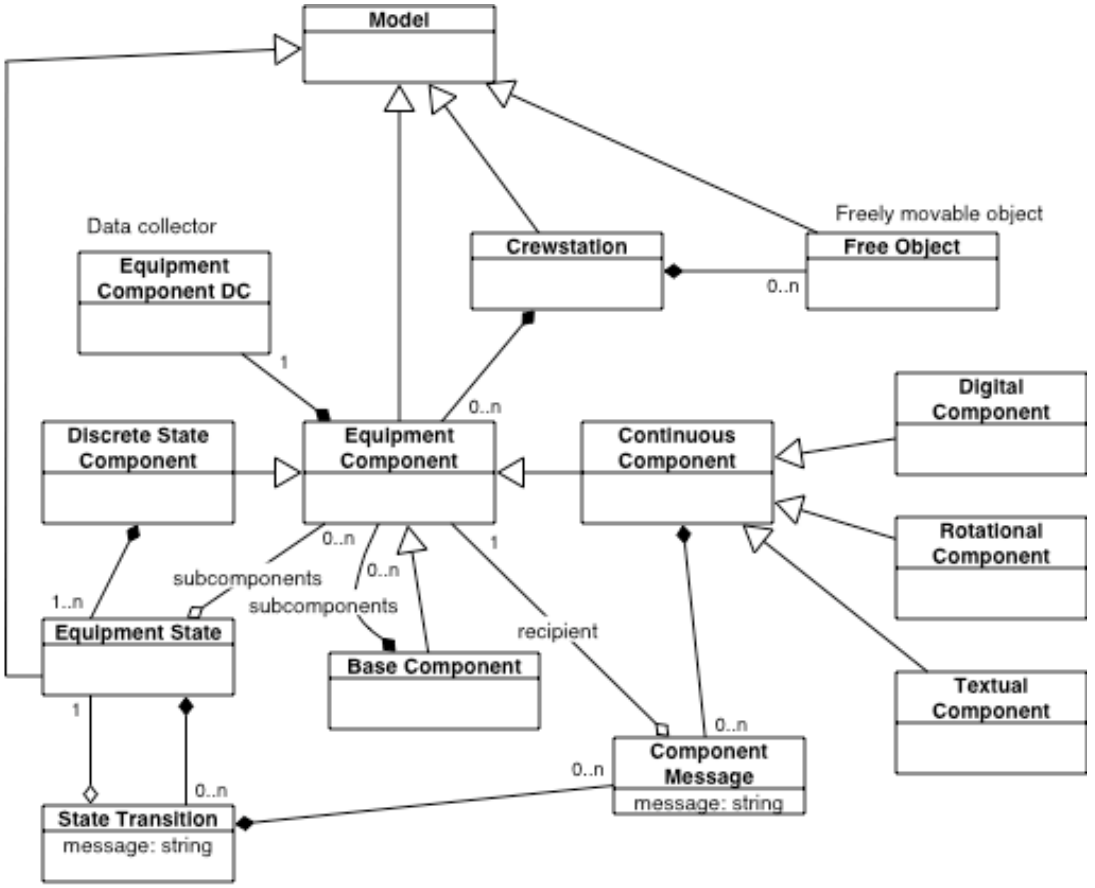
- Simulation definitions are a collection of models where models are operators, operator procedures, vehicles, crew stations, environments, events and situational awareness weightings.
  - E.g. To create a simulation, select the simulation created by default when the project is created
  - Look for User Simulation 1 in the Tree View under Simulations
  - In the property grid, replace the Name with “training simulation” or the name you would like for your Simulation
  - Set Connect to Apex to True
  - Set Simulation End Time to X seconds (X = the length of the desired simulation)

MIDAS 4.0 - External Connection

- Is to open the communication ports to jack and Apex
- Expand the External Connections by clicking on the + to the left of it in the Tree View
- Set up the Apex Engine
- In the Properties window, the IPAddress should read 127.0.0.1 (this is your local host since Apex was started on your computer when you ran MIDAS)
- The IPPort should read 5557
- Set up the Jack Cport
- In the Properties window, type the JACK IP Addresses noted above into the textbox to the right of IPAddress
- Type the JACK Port noted above into the textbox to the right of IPPort

MIDAS - Crewstation Fundamentals

- Diagrammatic representation of the relation among the architectural components in MIDAS



Notes:
- Digital, Rotational, and Textual components each have an unused ComponentMsg member that is redundant and can be removed.
- Visualization of the crewstation is provided through the Model class. A key feature is that when a Discrete State component changes state, only the geometry for that state, and of its subcomponents, are visible.
- Perception of the crewstion is provided through the Model class. A key feature is that for any equipment component, the attributes perceivable at any given moment are those attached to the component, plus those of any subcomponents. For a Discrete State component, only the attributes of the current state (plus its subcomponents) are perceivable.
- This model was designed to support complex MFD's. An MFD is essentially a Discrete State Component whose states are pages.

MIDAS 4.0 - Crewstation list

- Lists a number of attributes (crewstations) in the file structure to allow the user to re-use crewstations developed in previous applications
- This is where you insert crewstation components into the simulation
  - Discrete state components (states, state definitions)
    - Definitions accessed primarily through the MIDAS Properties contextual window - Attributes, Component states, Geometry (animation names, bounding box, initial position/rotation), Identification, Notes
  - Base components
  - Continuous/rotational components

MIDAS 4.0 - Crewstation Details

Things to Remember

- The Animation field (name) - Jack name of the object
  - Corresponds to the name used in the CAD/animation program
  - Links the animation from Jack (or any outside CAD environment) with the procedure development within MIDAS
- The Identification field - the MIDAS name
- Assigning scan patterns (and creating them if they do not exist)
  - The scan pattern names have to be user defined
    - Creating a scan pattern:
    - Look under the crewstation definition for internal scans, use ellipses under perception area to add to scan patterns
    - Assign the equipment components in the scan (shift click to enable multiple selection; control click for non contiguous components)
    - Give each pattern a name
    - In Apex Author, the argument drop down when you select scan with pattern primitive will be filled with the patterns you've already populated and named
- Crewstation list supports the state, size, attributes that drive the perception and motor control model

MIDAS 4.0 - Environment List

- List of possible MIDAS environments
- Can trade off lighting and visibility
- Environments themselves directly support the external perception model

MIDAS 4.0 - Event Set List

- Events are used to trigger the onset of behaviors
- Events can satisfy DSC dependencies
- Event sets are collections of related events that are used for specific simulations
- Events and event sets support context changes
- Modeler should select one event set per simulation, modeler can use event sets to tests varying equipment behavior (steam generators and time to diagnose failure)
- E.g. event set that has behaviors that mimicked a large steam generator versus those reflective of a small steam generator (primarily in terms of time)

MIDAS 4.0 - Operator List

- The MIDAS Operator list stores MIDAS details associated with an operator that has been imported from Jack
- MIDAS primitives drive the operator
- Allows the MIDAS modeler to customize one of the Jack character names to something more reflective of the model being created in MIDAS
- Attribute, body, error, execution, geometry, identification, models, and notes

MIDAS 4.0 - Operator Model

- E.g. Select Operator List in the Tree ViewRight Click and select Add Operator from the context menu
- Type “Joe” in the Name field
- Click on the + beside Effectors to see the location of all the effector positions that have been imported
- Can modify the effector positions to reflect starting position for the operator
- Select the initial Apex procedure or no behaviors will happen (but events will still happen as events are tied to the simulation)
- Operator Definition is also where the initial SA context is defined

MIDAS 4.0 - Vehicle Lists

- Defines the vehicles that are to be included in the simulation
- Modeler definition of vehicles to be included in the simulation
- Vehicle list in MIDAS is tied to the CAD environment
- Vehicles are named in MIDAS in a manner similar to the way it is done in the crewstation
- Waypoints can be imported from Jack
- Can be re-used in new simulations
- Attribute, body, error, execution, geometry, identification, models, and notes

MIDAS 4.0 - Situational Awareness List

- Awareness of one's environment
- SA list contains components important for SA for each context
- 3 levels of SA perception (internal)
  - Visual - Check-read
  - Visual - Exact-read
  - Auditory - Detection
  - Auditory - Comprehension

MIDAS 4.0 - Situation Awareness Model

- Add context, categories and situational elements (SEs)
- Situational Elements
  - Defining the aspects of the environment to which the operator brings into their awareness
  - Start using this from the bottom up by adding the Ses
- Categories
  - Used to represent groupings of SE's, and categories are weighted by context
  - Attributes, ID, models, notes
- Context
  - Used to define scenario phase of importance of Ses
  - Represent groupings of monitor categories, and weight the monitor category a full value of 1

MIDAS 4.0 - Apex Procedure List

- Behaviors coded here drive the operator procedures in the simulation
  - E.g. In the Apex Author tab, Right Click and select Add Procedure from the context menu
  - Type “Monitor” in the Name field
  - Select Joe from the Operator area of the Tree View
  - Select “Monitor” as the Initial Apex Procedure
  - Return to the Apex Author tab, Double Click on the Monitor procedure
  - Right Click and select Add Step or perception or attribute test

MIDAS 4.0 - Model Interaction

- MIDAS 4.0 uses Jack as the CAD GUI
- Communication can exist between equipment components (this functionality existed in previous versions of MIDAS)
- In Jack, click on properties and make the object visible/invisible depending on the state of the equipment component you are modeling
- Power behind MIDAS is that swapping components drive attribute values thus driving the perception model

Perception Tests

- Dependencies - “wait for” conditions
- Used when you want to test for dependencies on from visual/auditory perception
- Time to perception level dependency driven by the attributes
- Scan patterns constrain perception

Attribute Tests

- When you want to test for **dependencies** between elements
- Serves as the trigger for an activity
- Dependencies are satisfied when the simulated operator perceives an event, not when the event occurs
  - If the dependency is met, then the relationship will be true.
  - If the dependency is not met, the relationship will be false and the task will remain unmet and will not trigger

MIDAS models

- Perception
  - Vision
  - Audition
- Attention
  - Multi-resource model
- Motor Model
- Memory Models
  - STM/LTM Structures (beliefs/facts)
- Workload
- Situation Awareness

MIDAS Models: Visual Perception

- External
  - Peripheral - 160 degrees
  - Extended Foveal - 20 degrees
  - Foveal - 2.5 degrees
  - Perception level - f(dwell time, perceivability)
  - Perceivability - f(visibility, size, distance, contrast ratio)
  - Levels of perception - detection, recognition, identification
- Internal
  - check and exact read are the perception levels
  - Attribute type drives time to detection (ie digital, symbolic or text have different times to perception)

MIDAS Models: Vision

- Operator perceives visual objects through the attributes that are attached to the objects
- FOV is divided into three zones: foveal (2.5), extended foveal (20); and peripheral (150)
- Foveal gives the operator specific information about an object and involves fixating the attention to the object
- Extended foveal is used for the scanning model
- Peripheral is used to capture visual events outside of foveal vision (e.g. warning light)

MIDAS Model: Vision

- Interior vision
  - Concerned with reading states/values/messages off of instruments (usually only requiring one attribute)
- Exterior vision
  - Stages - detection, recognition, and identification
  - Depends on two factors: dwell time and perceivability of object
  - Exterior scans follow a predetermined scan pattern on the window
  - Each scan point of the scan pattern is fixated in sequence
  - Objects in foveal area are fixated before moving to the next fixation point in the scan pattern
  - Dwell time is 300ms for detection, 1 second for recognition and 3 seconds for identification

MIDAS Model: Vision

- Detected
- Recognized
- Identified

- Size
  contrast
  movement
  location
- Class
  shape
  motion
- Texture
  color
  transparency
  ID

MIDAS Models: Vision

- Perceivability f(visibility, size, distance, local contrast ratio)
- Three stage model for computing probability of detection recognition and identification
  - Stage 1: Compute actual and apparent contrast
  - Stage 2: Find threshold contrast for target
  - Stage 3: Determine probabilities of detection, recognition and identification

MIDAS Models: Attribute 3 Types

- General (objects) attributes
  - ID, class, location orientation, motion, movement
- Visual Attributes Types:
  - Size, shape, texture, color, contrast, transparency, text, digital integer value, digital float value, spatial integer value, spatial float, light emitting, heat emitting
- Auditory attributes
  - Sound-emitting, signal type, loudness, pitch, speech

MIDAS Models: Vision - Events

- Visual events
  - Are changes in the environment that catch the operator's attention
  - Is perceived through the peripheral vision, causes the operator to fixate on the object (not currently implemented)
  - Is user defined
  - E.g. Fixates pre-empts scan pattern

MIDAS Models: Auditory Perception

- Two stages of processing
  - Detection
  - Comprehension
- Content
  - Verbal strings or signals
  - All or none processing assumption for “say-message”
  - Interrupts currently disrupts the entire message for speech
  - For auditory detection, both signals will be perceived
  - MIDAS workload for both fixate and listen to the workload is calculated from the types of attributes (e.g. signal versus speech)

MIDAS Model: Audition

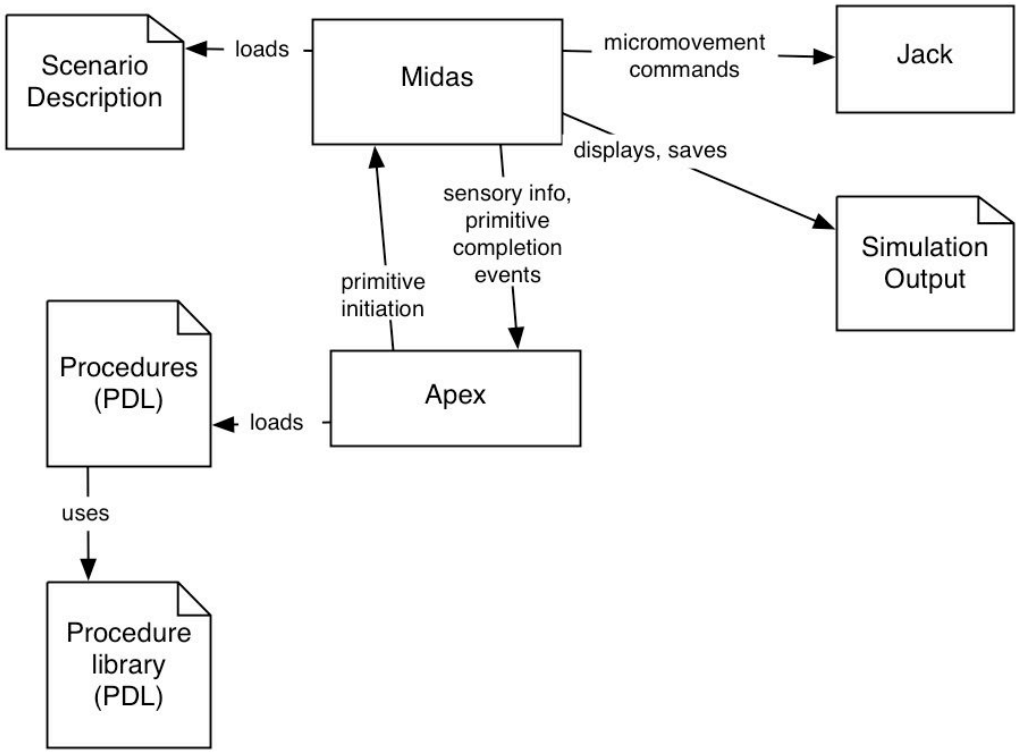
- Auditory attribute or events triggers the listen task
- Sounds need to be generated from within the crewstation for the operator to perceive them
- Speech is a simple text string
- Auditory perception occurs in two stages
  - Detection
  - Comprehension

MIDAS Models: Motor

- Manages physical effector positions and object location in XYZ space
- Leverages off of Jack's CAD environment and Jack's anthropometry
- Uses Fitt's Law for determining reach times
- Implements a micro model for fine motor tasks

Motor Model

- MIDAS simulation engine drives procedures, performance times, and Jack activities
- Jack completes the activity
- Apex engine sends/receives MIDAS performance times
- MIDAS executes next activity



Attention: Multi-Resource Model

- Task Analysis and Workload (TAWL) Index and the Modified TAWL McCracken and Aldrich (1984), Modified/Augmented by Mitchell (2000)
- Empirically determined scale of performance based on the human behavior of helicopter pilots
- 4-Channel, 7-point workload threshold/characterization
- Applied at the task primitive level
- Tracks attentional resources according to the 6-channels of visual/auditory encoding, spatial and verbal cognitive processing, and manual/voice output
- Implements a resource-conflict matrix to compute loads in each channel

Attention: Multi-Resource Model

- MIDAS instantaneous workload model

$$W_j = \sum_{i=1}^{i=6} (a_{t1,i} + a_{t2,i} + a_{tn,i}) * c_{i,j}$$

$W_j = instantaneous\ workload\ of\ channel\ j\ at\ time\ t$

$i, j = 1 \ldots 6\ interface\ channels$

$t1, t2 = operator\ tasks$

$a_{t,j} = load\ of\ channel\ i\ to\ perform\ task\ t$

$t_i = interface\ channel\ i\ associated\ with\ task\ t$

$c_{i,j} = conflict\ between\ channel\ i\ and\ j$

$where\ either\ a_{t1,i}\ or\ a_{t2,i}\ is\ non-zero$

MIDAS
man-machine integration design and analysis system

Attention: Multi-Resource Model

- Conflict matrix when multiple tasks require the same channel
- Adapted from W-Index (Riley, 1989)

Task load	Performance decrement
0.1-7	0%
7.1-8	10%
8.1-9	20%
9.1-10	30%
10.1-11	40%
11.1-12	50%
12.1-13	60%
13.1-14	70%
14.1-15	80%
15.1-16	90%
16.1-	Not permitted

	vis	aud	cogs	cogv	mout	vout
vis	1	0.3	0.2	0	0.2	0
aud	0.3	1	0	0.2	0	0.2
cogs	0.2	0	1	0.5	0.2	0
cogv	0	0.2	0.5	1	0	0.2
mout	0.2	0	0.2	0	1	0.3
vout	0	0.2	0	0.2	0.3	1

- Conflict matrix is applied, resultant workload affects performance level

MIDAS 4.0 - Workload Model

- To use primitive in MIDAS 4.0
  - Edit in the properties grid - you can operator procedures use primitives and the workload values are pulled from those primitives
  - This insures workload consistency and prediction
- Examine the output of the workload model to verify and perform validation of the model output

Situation Awareness Model

- Computes two quantities: actual SA and perceived SA.
  - Actual SA is the portion of situational elements that the operator knows relative to the situational elements that he would know in the ideal state
  - Perceived SA includes the situational elements for which the operator has no knowledge
  - The SA model is comprised of three key features:
    - 1) *situational elements* (SE),
    - 2) *categories* (context sensitive nodes), and
    - 3) a regulatory mechanism, the *SA manager*.
  - For each defined context a set of critically important SEs categorized into weighted groupings specify the ideal SA. When the operator's context changes due to new environmental conditions, the weightings on the categories change their values.

Situation Awareness Model

- 3 levels of SA perception (correspond to the visual perception)
  - Detection
  - Recognition
  - Identification
- No error SA - Perceived SA does not include error

SA Details: Situational Elements

- Objects external to the crewstation

Level	Characteristic	% SA
undetected	no info	0
detected	location	33
recognized	category	67
Identified / comprehended	specific type, contextual meaning	100

- Objects internal to the crewstation

Level	Characteristic	% SA
unread	no info	0
check-read	approximate value	50
read	exact value	100

SA Details: Situational Context

- Situational Elements/Categories
  - 4 categories defined
  - E.g. Two contexts defined: Attack and emergency landing

Category	Targets/Threads	Battlefield	Own-aircraft	Landing Areas
Situational Element	Tank-1	Road Junction	Radar	Corn Field
	Tank-2	Oak Tree Grove	Altimeter	Meadow
	Missile Launcher	Building-1		Road Junction
		Pine Tree Grove		

- Situational Context
  - Environment drives the context weights

Category	Context: Attack	Context: Emergency Landing
Targets/Threats	0.6	0.1
Battlefield	0.3	0.0
Own-aircraft	0.1	0.1
Landing Areas	0.0	0.8

Perceived/Actual SA Calculated

- Perception level and weights at specific instance of time during attack

Element	Node	Perception Level	Proportion SA
Tank-1	Targets/Threats	recognized	0.67
Tank-2	Targets/Threats	undetected	0.0
Missile Launcher	Targets/Threats	identification	1.0
Road Junction	Battlefield, Landing Area	identification	1.0
Oak Tree Grove	Battlefield	recognized	0.67
Building-1	Battlefield	detected	0.33
Pine Tree Grove	Battlefield	identification	1.0
Corn Field	Landing Areas	detected	0.33
Meadow	Landing Areas	detected	0.33
Radar	Own-aircraft	check-read	0.5
Altimeter	Own-aircraft	read	1.0

- Perception level and weights at specific instance of time
- Perceived SA calculated by taking weighted average for each known SE in a category of SA
- Actual SA calculated by taking an average of all SE possible in an instance (including SE=0)
- Currently you can define your own categories and contexts

NASA-Ames Research Center

Human Systems Integration

MIDAS - INL Training Day 2
Additional Model Features

Thursday October 12, 2006

San José State UNIVERSITY

MIDAS
man-machine integration design and analysis system

Adding Behavioral Models

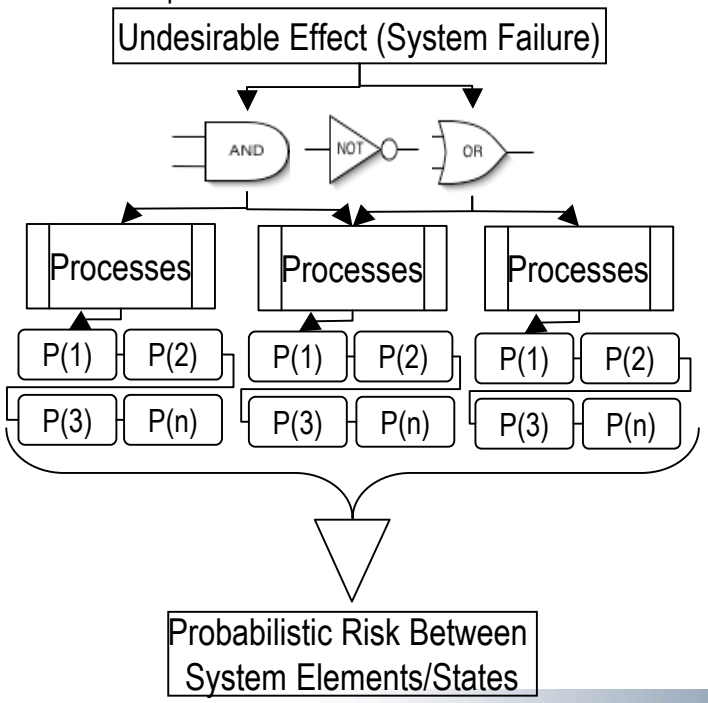
- Motor control model (C-Sharp, can be parameterized but not user-edited)
- Timing - code the time estimate function (Fitt's Law)
- Assign the workload/attention (primitives.xml)
- Consider whether you have any impact on the perception model (ie STM)
- Jack interactions (and the MIDAS simulation clock)

Stochasticity and Variability

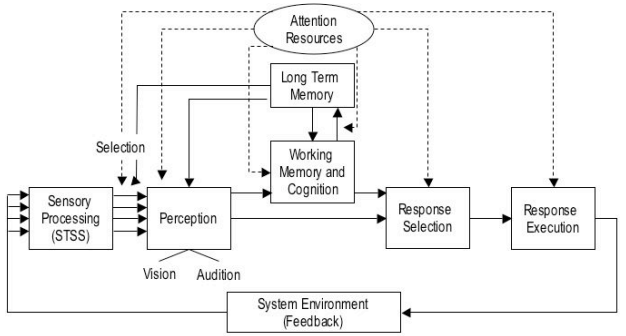
- Future Efforts
  - Multiple operators
  - Feedback from Jack - 3D environment
    - Reach and Push and Release success/failure
    - Effector location updates during simulation
  - Variability - introduce it on the timing of the events, the MIDAS primitives, selecting procedure path selection (decision making based on path logic), multiple runs, output is limited in terms of spread (currently tunnelled)

Fault Tree Analysis and Modeling

- FTA is a system development technique to analyze system reliability and safety prior to deployment of large scale, highly complex systems
  - Provides valuable insights/predictions into system risks (e.g. per thousands of hours of operation)
  - Can be static (paper based/analytical) or dynamic (computational)
  - Does not consider human risks to system performance



- Modeling is a technique to better understand the human contribution to system risks by examining human performance characteristics (cognitive or physical) that in turn increases the system's vulnerability to failure through error prediction
- Operates on a different scale than FTA (smaller number of hours of operation)



- Similarities exist between the two approaches
  - FTA can identify system risks, human performance models can identify the human vulnerabilities (risks)
  - Task as the behavioural unit, common assumptions (interdependence of tasks/ task performance), modification of task performance probabilities by PIF
- Work is being considered to link the approaches